JOURNAL OF AERONAUTICS AND SPACE TECHNOLOGIES
(ISSN: 1304-0448)
January 2020 Volume 13 Number 1
www.jast.hho.edu.tr

Research Survey

Systematic Reviews in Model-Driven Engineering: A Tertiary Study

Deniz AKDUR[1], Onur DEMİRÖRS [2]

[1] *ASELSAN Inc., Ankara, Turkey*, denizakdur@aselsan.com.tr, https://orcid.org/0000-0001-8966-2649

[2] *İzmir Institute of Technology, Computer Engineering Department, İzmir, Turkey*, onurdemirors@iyte.edu.tr, https://orcid.org/0000-0001-6601-3937

Article Info

Received: *September 9, 2019*
Accepted: *December 3, 2019*
Online: *January 23, 2020*

Keywords: Systematic mapping (SM), systematic literature review (SLR), model-driven engineering (MDE), software modeling, modeling characteristics

Abstract

To cope with growing complexity of software-intensive systems, model-driven engineering (MDE) has become a widely used approach in the industry by providing many (potential) benefits with different purposes. Although there has been an increasing interest in conducting secondary studies among MDE researchers such as surveys, systematic mapping (SM) and systematic literature review (SLR), there have been no tertiary study to synthesize the findings from all these existing secondary studies, which also examines various characteristics of software modeling (e.g., purposes, benefits and challenges) as a meta-analysis. The objective of this paper is to investigate and understand the state-of-the-practices in MDE based on the modeling characteristics by presenting a tertiary study (i.e., a systematic review of systematic reviews). To this end, we collected the set of all the existing 64 secondary studies in this field using a well-defined search strategy. This article presents inputs for different modeling stakeholders to better understand and use different purposes, benefits, and challenges of MDE by aggregating consolidated findings on this approach.

To Cite This Article: D. Akdur, O. Demirors, "Systematic Reviews in Model-Driven Engineering: A Tertiary Study", Journal of Aeronautics and Space Technologies, Vol. 13, No. 1, pp. 57-68, Jan. 2020.

Model-güdümlü Mühendislik Üstüne Sistematik Gözden Geçirmeler: Bir Üçüncül Çalışma

Makale Bilgisi

Geliş: *9 Eylül 2019*
Kabul: *3 Aralık 2019*
Yayın: *23 Ocak 2020*

Anahtar Kelimeler: Sistematik haritalama (SH), sistematik litaratür taraması (SLT), model-güdümlü mühendislik (MGM), yazılım modellemesi, modelleme karakteristikleri

Öz

Yazılım-yoğun sistemlerin artan karmaşıklığı ile başa çıkmakta bir araç olarak görülen model-güdümlü mühendislik (MGM), farklı amaçlar için sağladığı kazanımlarla endüstride yaygın olarak kullanılmaya başlanmıştır. MGM araştırmacıları arasında anket, sistematik haritalama (SH) ve sistematik literatür taraması (SLT) gibi ikincil çalışmaların yapılmasına ilgi artmasına rağmen, tüm bu çalışmalardan elde edilen bulguları sentezleyen ve yazılım modelleme karakteristiklerine göre ayrıştıran (örneğin, modelleme amacı, faydası ve karşılaşılan zorluklar gibi) bir üçüncül çalışma bulunmamaktadır. Literatürdeki boşluğu dolduran bu çalışmada, MGM'deki uygulama durumlarını yansıtan mevcut 64 ikincil çalışma, modelleme karakteristiklerine göre gruplandırılmıştır. Sistematik inceleme çalışmalarının sistematik çalışmasını sunan bu makale, MGM'nin farklı amaç, fayda ve zorluklarını daha iyi anlamaları için tüm modelleme paydaşlarına (araştırmacılar da dahil) girdi sağlayarak, değişik MGM kullanımlarına modelleme karakteristikleri bazlı olarak ışık tutacaktır.

1. INTRODUCTION

Model-driven engineering (MDE), which is considered as one of the most popular approaches in software modeling, provides different software engineering (SE) roles (e.g., from developers to testers and systems engineers) the ability to abstract out details and helps to automate software development life cycle (SDLC) artifacts [1, 2]. To cope with growing complexity of software-intensive systems and due to economic factors such as fast time-to-market, MDE is seemed to be a reliable development process with short cycle times without accidental complexities [3]. Many practitioners in the defense and aerospace industry have started to adopt MDE and have benefitted from the models to support various SE activities not only in development but also in testing, and maintenance phases of SDLC besides during software certification processes [4-6]. However, software modeling and MDE practices vary since the characteristics of software modeling (e.g., purpose, benefit, challenge, stakeholder profile, SDLC phase, etc.) differ among systems as well as among sectors [7]. On one hand, some stakeholders (e.g., systems engineers) use software modeling and MDE to communicate with colleagues or to document analysis activities (e.g., requirement). On the other hand, other stakeholders use model-driven concepts with automated generation of code or documentation. It is also frequently observed that different units within the same company might use different modeling approaches for different purposes in different phases of SDLC with different benefits and challenges [8]. Despite the necessity of understanding different characteristics of software modeling in different context with a variety of MDE surveys (e.g., in [9-11]), it may be hard to locate consolidated evidences on MDE in several different publications; hence, it is important to systematically synthesize them.

Survey, systematic mapping (SM) and systematic literature review (SLR) are the established methods for classification and synthesis of scientific studies [12]. These "secondary studies" (i.e., studies of studies) play an important role both in supporting further research efforts and also in providing information to assist both researchers and practitioners [13]. On the other hand, a tertiary study aims to provide further information about the published secondary studies on a specific topic by tabulating their information. In the SE literature, there is only one tertiary study in MDE [14] that only focused on the quality factors, which is one of the benefits of MDE (i.e., quality improvements in [7]). Other fundamental characteristics of software modeling within secondary studies have not been synthesized in the literature (e.g., as a meta-analysis).

To bridge the gap in the existing literature, the objective of this study is to present different characteristics of software modeling (e.g., not only on one of the benefits as in [14], i.e., quality) in the secondary studies by synthesizing a wider picture of the empirical research in MDE domain. Through a systematic search, we have identified 64 papers published until 2019, whose software modeling characteristics (e.g., purposes, benefits and challenges) are varying. The consolidated findings of this study, which reflects the state-of-the-practices of MDE, present empirical evidence for all MDE stakeholders (e.g., practitioners, researchers and educators) to facilitate their work and their decisions by providing an insight while modeling. Moreover, by identifying unexplored research directions, this study is also beneficial for any researcher in selecting open areas of MDE.

The remainder of this paper is structured as follows. Section 2 gives the research methodology. Section 3 presents the overall results. Section 4 discusses the limitation and threats to validity. Finally, Section 5 concludes this study.

2. RESEARCH METHODOLOGY

To plan and conduct our literature review, we utilized one of the popular systematic review guidelines [15]. We next present the main phases of this systematic review, which will guide any stakeholder (either practitioner or academician) for MDE adoption as a meta-analysis.

2.1. Planning and Research Questions

During the planning phase, a formal protocol containing the details of the strategies for search and selection process, quality assessment, data extraction, data synthesis and data analysis was developed [12]. In order to achieve the goal, this tertiary study raised the following research questions (RQs):

- **RQ1:** Which purposes of MDE have been addressed in existing secondary studies?
- **RQ2:** Which (potential) benefits of MDE have been addressed?
- **RQ3:** Which (potential) challenges of MDE have been addressed?

The search strategy is presented in Table 1. During the search process, four digital libraries (i.e., Science Direct, IEEE Xplore, Google Scholar and Elsevier Scopus) were used. Besides automated searches in these digital libraries, to triangulate the results, manual search on referenced articles and personal web pages were performed as snowball sampling techniques [16]. This sampling strategy provided substantial new works as in our earlier studies [17].

Table 1. Search strategy.

Databases searched	Search Engines (Science Direct, IEEE Xplore, Google Scholar, Elsevier Scopus)
Manual search	Besides automated searches in digital libraries, manual searches on referenced articles and web pages are performed.
Search strings	(*model driven* OR *model-driven* OR *MDE* OR *UML* OR *DSL* OR *DSML*) AND (*systematic mapping* OR *SM* OR *systematic review* OR *literature review* OR *SLR* OR *survey*)
Topic Restriction	Software + Computer Science
Search applied to	Metadata only (Abstract/Summary & Title Text and Indexing Terms/Keywords) - if not possible, full text was searched
Language	Papers written in English
Publication period	until 2019 (exclusive)

While determining prior search strings, there were two major search groups in our study: "*MDE related strings*" AND "*secondary studies related strings*". In this prior set, there was no "*UML*" since using unified modeling language (UML) cannot guarantee model-driven techniques (i.e., some practitioners might use UML as a sketch or model-based approach without MDE [7, 18]). However, after conducting a pilot testing on the major search terms, it was realized that without "UML" in the search strings, there is a possibility to miss some model-driven approaches. Since the search was applied to metadata only, we could not find out some model-driven related papers, which include "automatic code generation" or "model transformation" (e.g., as in [19]). Then, it was decided to include "UML" in the search strings. But after finding such secondary studies, the full text was also searched whether there was a "model-driven" specific activities (e.g., code or document generation). Therefore, the final map of this study also fulfilled this selection criteria (Note that the search string was customized for different databases according to their interface requirements while keeping the logical order consistent).

Note that the preliminary version of this study was initiated when the need for understanding the state-of-the-practice of MDE in the embedded software industry was arisen to be used for a world-wide survey design [20] (Notice that all these preliminary findings were extensively used as inputs in [18, 21]). Then, for the study reported in this article, the publication period was extended to "until 2019".

2.2. Execution of the Systematic Search Process

After using search strings on selected digital libraries and performing manual searches if necessary, there were potentially 3527 relevant papers. Then, by applying exclusion/inclusion criteria, also removing duplicates and manually removing "personal opinion survey" papers, there were 64 papers in the final pool for attribute identification. All these processes is depicted in Figure 1.

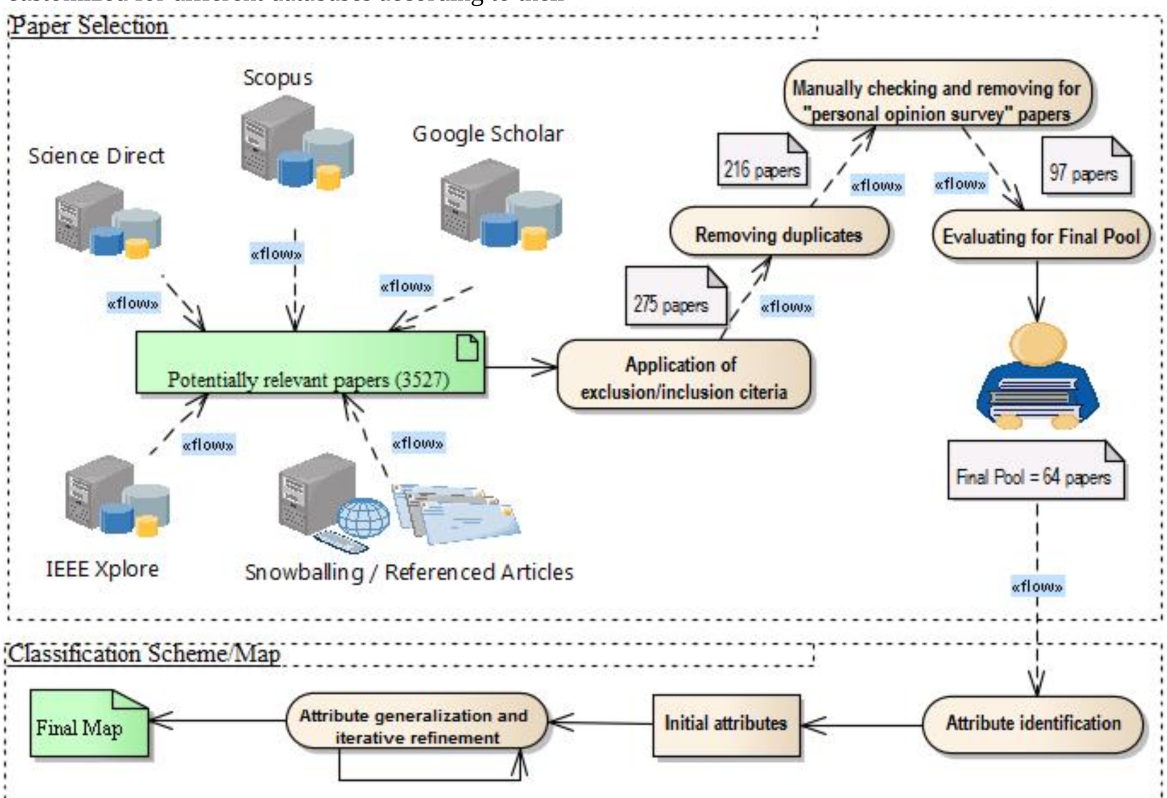
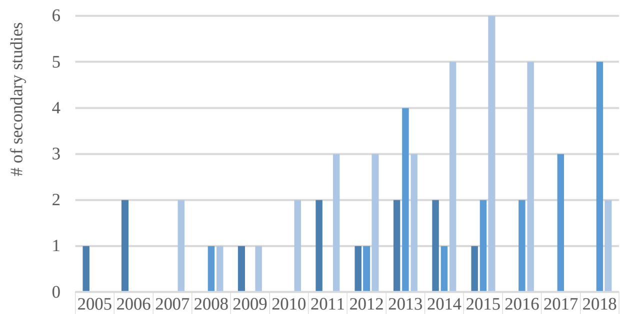


Figure 1. Tertiary study search process and its final map.

3. RESULTS AND DISCUSSION

The annual publication chart of the pool of 64 secondary studies is shown in Figure 2. 12 of studies under review are survey, 19 of them are SM and 33 of them are SLR. The final pool of sources and the online data extraction sheet can be found in [22] so that the reader might access all details of these secondary studies if s/he wants to explore them in depth.



	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018
Survey	1	2			1		2	1	2	2	1			
SM				1				1	4	1	2	2	3	5
SLR			2	1	1	2	3	3	3	5	6	5		2

Figure 2. Yearly distribution of the resulting secondary studies.

Note that to address RQs, since there were different terminologies to indicate the same purpose, benefit or challenge in different secondary studies, to get a common language and catalogue, similar items were combined in a single item, which we will present next (e.g., high abstraction level and understanding a problem at an abstract level are the same purpose; or cost savings or reduce costs are the same benefit and they were all combined in a single item).

3.1. RQ1: Purposes of MDE

Purpose of the modeling and modeling rigor (e.g., the degree of modeling formality), which are two main characteristics for software modeling are strongly related with the medium type used, which is another characteristic of software modeling [7]. Moreover, SDLC phases, where modeling is used (e.g., analysis, design, or implementation) are also affected by the modeling purpose [7]. In other words, the "purpose" of modeling affect the other software modeling characteristics. Therefore, it is necessary to understand the different purposes of MDE, which treats software modeling artifacts (e.g., prescriptive diagrams) as the primary artifact of the all SE process such as development, testing and maintenance [1].

There are 58 secondary studies, which explicitly mention about the purposes of MDE (i.e., 90% of final map). When we generate a derived set by grouping similar purposes, the result is given in Table 2 (See [22] for all data).

The purpose of modeling can be decomposed into two groups: general modeling purposes and MDE-specific purposes [7]. "Communication", "Understanding a problem at an abstract level" and Documenting analysis and design" are general and common purposes of software modeling for both descriptive and prescriptive modeling [7]. For example, the modeling stakeholder, whose approach is sketching or model-driven might have these general modeling purposes. However, "Code generation", "Documentation generation", "Model-based testing (MBT) / Test case generation", "Model transformation" and "Model simulation" are only specific to the model-driven approach, which is prescriptive modeling [18].

Table 2. Purposes reported in secondary studies.

General Modeling Purposes	Understanding a problem at an abstract level
	Communication between stakeholders
	Documenting analysis & design
MDE-specific Purposes	Code generation
	Documentation generation
	Test case generation / Model-based testing (MBT)
	Model simulation
	Model transformation

The distribution of the purposes based on individual secondary studies is depicted in Figure 3 (Note that PIDs stands for "Paper ID"s, which is given in APPENDIX – Final systematic mapping). As seen, the most reported purposes are "Understanding" and "Model transformation". Note that "Model transformation" in MDE is an automated way of modifying and creating models. This might be occurred as Model-to-Model (M2M), Model-to-Text (M2T) or Text-to-Model (T2M). In fact, "Code generation" includes a "model transformation", but in this context, we searched for any string related with "model transformation" explicitly even if the secondary study includes "code generation" or "documentation generation" (e.g., some papers include "code generation" but does not include any "model transformation" concept in the text explicitly such as PID11, PID22, PID28).

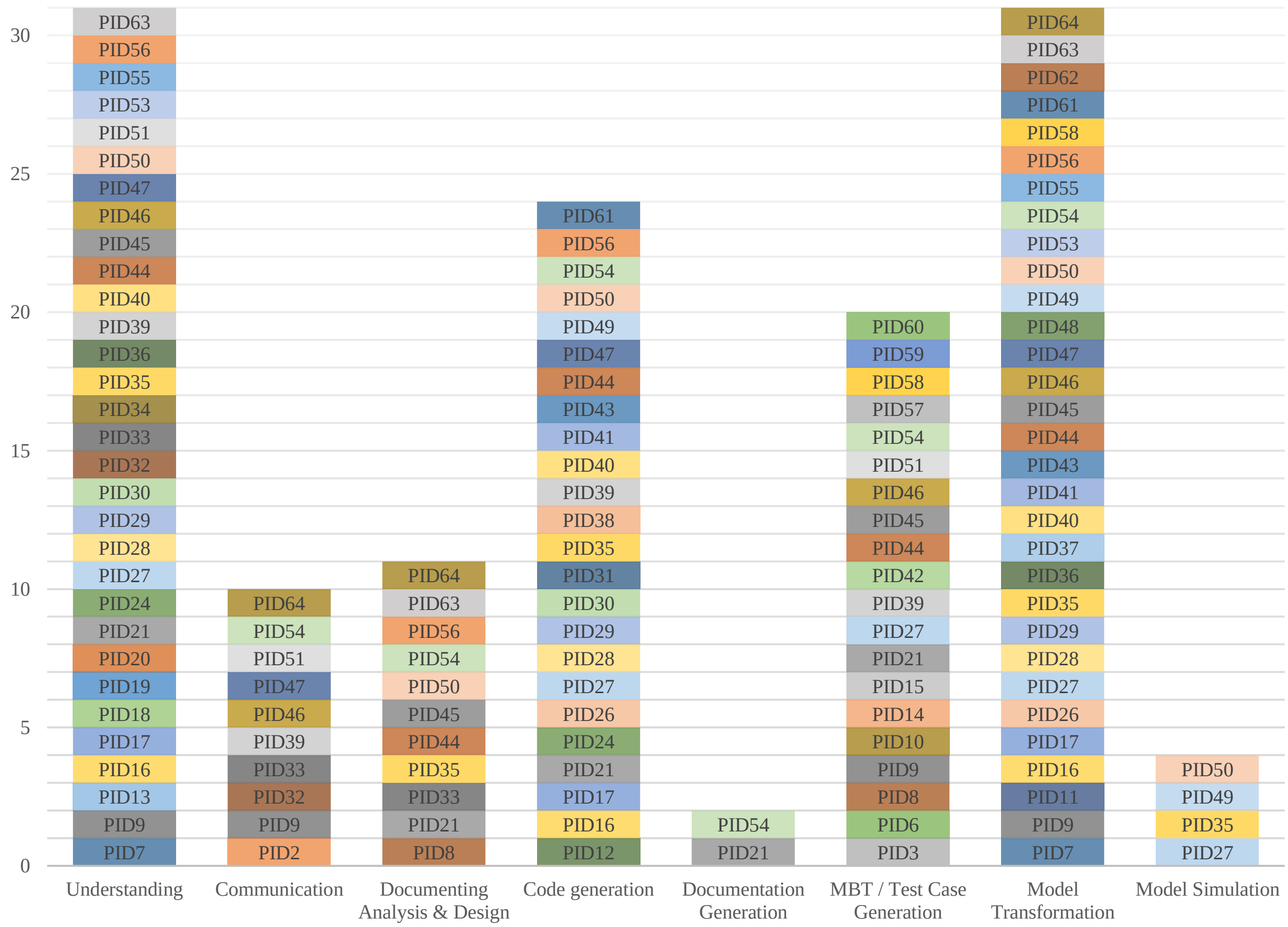


Figure 3. Modeling purposes distribution of resulting secondary studies.

3.2. RQ2: Benefits of MDE

Using MDE provides different types of benefits for different modeling stakeholders such as software developer, software tester or systems engineer, who has different purposes (e.g., either for general or MDE-specific purposes).

There are 56 secondary studies, which explicitly mention about the benefits of MDE (i.e., 87.5% of final map). The reported benefit set is given in Table 3.

The distribution of these benefits based on secondary studies is depicted in Figure 4. The most reported benefits are “quality improvements”, “manage complexity”, “cost savings” and “shorter development time”. As in any engineering activity, software development projects should also be completed within anticipated budget (cost), within anticipated schedule (time) with conformance to requirements (quality) [23]. This viewpoint is also conformant with the results depicted in Figure 4.

Table 3. Benefits reported in secondary studies.

Cost savings & reduce costs
Shorter development time & time to market & time reduction
Quality improvement & High quality code
Reusability
Maintainability
Extensibility & Expandability
Portability
Productivity
Traceability
Reliability
Interoperability & Integrity
Ensuring SDLC artifact (e.g., source code, documentation, test driver, etc.) and model compatibility & Reduced number of accidental programming errors
Manage complexity & Understandability
Team collaboration & Communication
Test effectiveness & Guaranteeing the verification of important properties of a system in the early stages of SDLC

Note that all individual quality factors (e.g., reusability, maintainability, productivity, portability, etc.) leads “quality improvements and high quality code”. However, in this study, we looked for “explicit” usage of “quality improvements” related strings although some secondary study might include individual quality attributes (e.g., PID4, which mentions about reusability and maintainability, but does not include any “quality” related string). On the other hand, some studies also mentioned about “quality improvements” without explicitly mentioning these individual quality factors such as PID2, PID3, PID7, and PID16. Therefore, we decided to separate these groups although individual factors affect having high quality code. Note that in that case (e.g., if we counted individual quality factors such as maintainability and reusability as an indirect improvement factor into software quality) “quality improvements” count in Figure 4 would be increased.

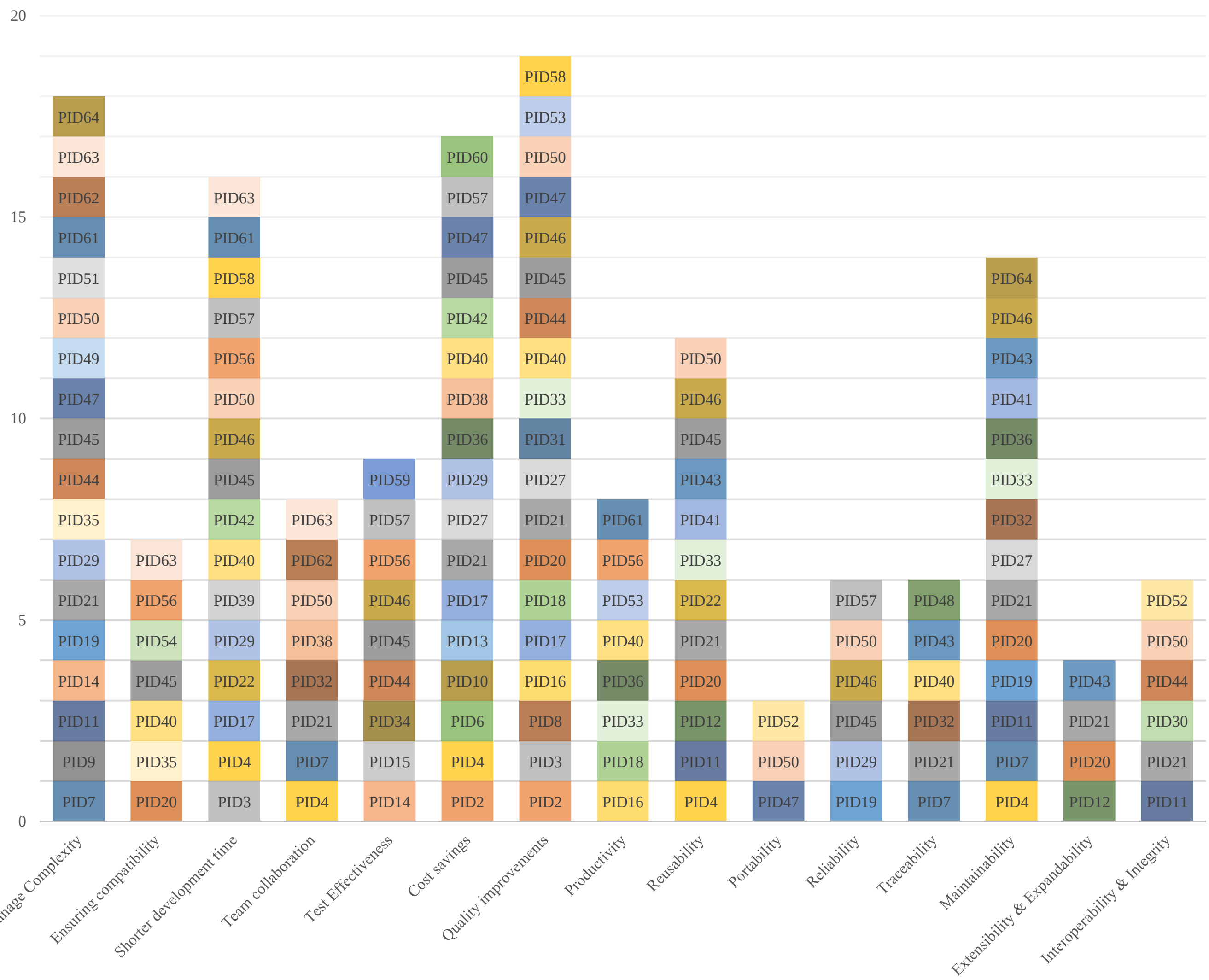


Figure 4. Modeling benefits distribution of resulting secondary studies.

As in the case of purposes, there are MDE-specific benefits and general benefits of software modeling [7] and this is also related with the software modeling usage approach (e.g., sketch, model-based, model-driven approaches). “Managing complexity”, “Team collaboration” are all treated as general benefits of software modeling (e.g., the stakeholder might use pen and paper to understand a problem at an abstract level as sketching and achieve to manage the complexity of the system). However, since there is an automatic generation of artifact (e.g., code, documentation, test case), “Ensuring compatibility” is mostly achieved in model-driven approach, which is also related with one of the other characteristics of software modeling, correspondence [7].

3.3. RQ3: Challenges of MDE

There are 45 secondary studies, which explicitly mention about the challenges of MDE (i.e., 70% of final map). The derived challenge set is given in Table 4.

While mentioning about tool support problems, these secondary studies pointed out specific challenges related to tools (e.g., Back/Forward compatibility issues between tool versions, difficulties in taking technical support from the tool supplier, difficulties with code generation capabilities, difficulties with model level debugging, difficulties with traceability support, difficulties with version management support, high effort for training, lack of model checking capabilities and many usability issues in its editor).

The distribution of these challenges showed that the most frequently reported MDE challenges are "tool support", "modeling languages itself" (such as DSL and UML problems) and "model transformation & merging" challenges as depicted in Figure 5.

Table 4. Challenges reported in secondary studies.

Tool support
Model quality (e.g., how to define, assure, predict, measure, improve and manage it?)
Model verification/validation techniques
Modeling expertize in the company
Modeling languages itself (e.g., domain specific language (DSL) problems such as how to define concrete & extensible DSLs in different context; and UML semantics and syntax problems)
Automatic artifact generation problems (code, test driver cases) & Optimization and performance issues with automatic code generation & Reverse engineering
Model refactoring
Training
Transformation/merging of models (e.g., how to integrate/merge models in different projects?)
Organizational resistance to change & Understanding and acceptance of the model driven concept & Lack of process for MDE

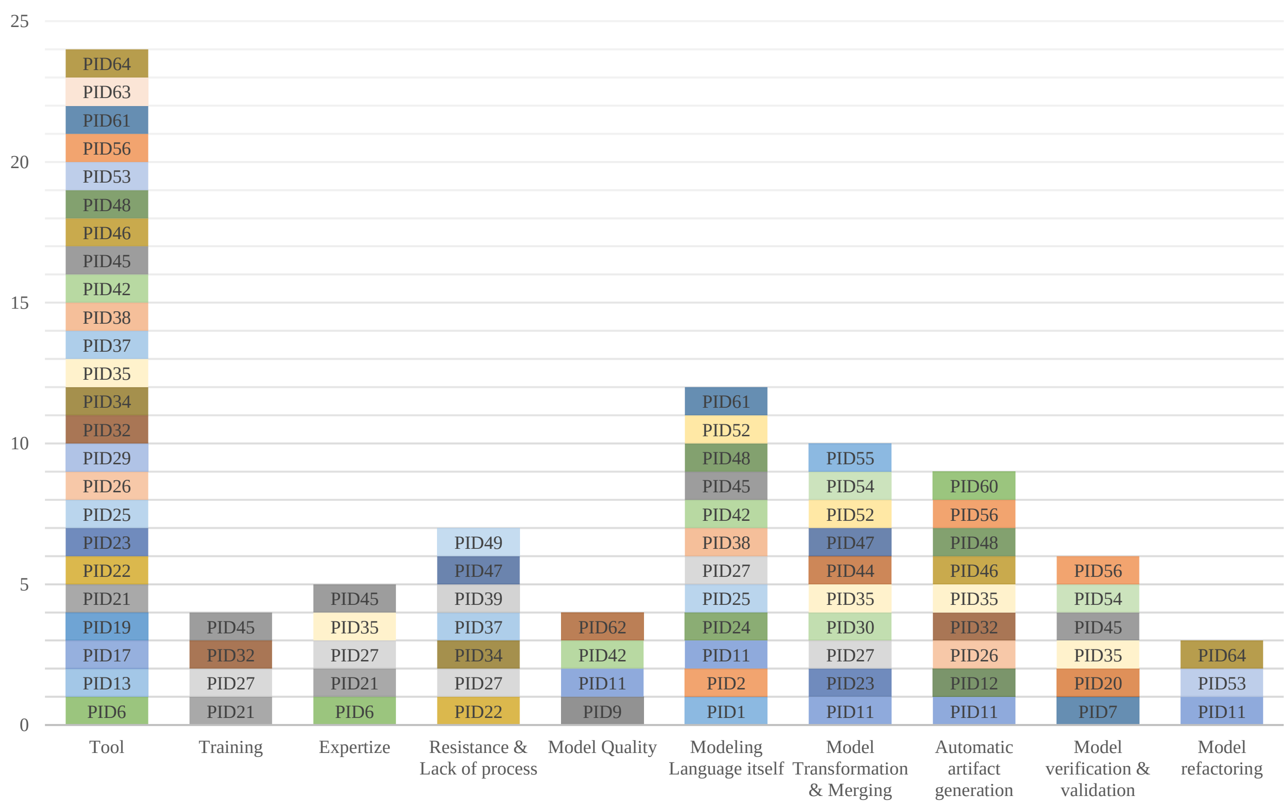


Figure 5. Modeling challenges distribution of resulting secondary studies.

Modeling challenges can be categorized into two groups: "Organizational" and "Technical" [7]. Note that tool, training, expertize and resistence (e.g., the first four challenges in Figure 5 from the left side) are organizational challenges; whereas the remaining ones (e.g., model quality, modeling languages itself, etc.) are technical challenges. The results showed that although the most frequently reported challenge is an organizational one (i.e., tool), the other top three challenges are technical, which might lead the researchers into these open problems.

Apart from these three RQs, in [22], other attributes of these secondary studies such as keywording, RQ(s) and RQ types, etc., are presented for the ones, who are interested.

4. LIMITATIONS and THREATS to VALIDITY

In this section, we discuss the limitations and possible validity concerns in our study, which we minimized or mitigated.

A threat to the validity of our study is the selection of the used digital libraries and the search terms. We addressed this threat by using four of the most important digital libraries in computer science that allowed us to define accurate search terms by providing a versatile set of possible search query constructs.

Although we follow the search strategy based on the guidelines of evidence-based software engineering (EBSE) (e.g., [12, 15]) to ensure the completeness of our sample, there is always a risk that there would be some papers that were not included in our final pool due to their unavailability in digital databases searched or because they may not have used the relevant keywords in their metadata (e.g., title or abstract). To mitigate this risk, we attempted to improve our results with secondary searches (as in the case of "including UML" in the search string) and by using snowballing [16].

While generating derived set and creating common catalogue for purposes, benefits and challenges of MDE, it is possible that other researches may select different categorization (e.g., terminology) and their derived set might be different.

Another threat to the validity of our results is, of course, that many practical experiences on purposes, benefits and challenges of MDE do not get published in any secondary studies, because they are made by practitioners, who might not want to write scientific papers. To mitigate this, we followed another research strategy (i.e., the opinion survey [20]), which utilized the results of this systematic review study as a baseline.

5. CONCLUSION

The tertiary study reported in this paper has identified 64 secondary studies until 2019 (exclusive) about MDE. This paper represents the first ever tertiary study in MDE research literature to understand software modeling characteristics related to MDE in detail (e.g., purpose(s), benefit(s) and challenge(s)).

The results showed that the most reported purposes of MDE are "Understanding a problem at an abstract level", "Model transformation" and "Code generation". According to the results, "quality improvements", "manage complexity", "cost savings" and "shorter development time" are the most reported benefits. On the other hand, the results revealed that the most frequently reported MDE challenge is "tool support", which is a mandatory concept for all model-driven approaches.

We believe that the findings of this systematic review study would be useful both MDE researchers and practitioners since it provides an effective starting point for an overview of all the researched areas so far.

We have been able to identify research gaps (opportunities and threats for MDE [22] such as model quality, adoption, empirical evaluation of model driven approaches in the industrial environment) from our mapping that is also beneficial for those interested (e.g., both researchers and practitioners) in selecting fruitful open research areas of MDE.

We would like to study technical and social factors that influence the adoption of MDE. We plan to conduct a systematic review on this topic first; and then case studies in the industry to better analyze these factors.

6. ACKNOWLEDGEMENTS

The authors would like to thank Dr. Vahid Garousi, who contributed to the initial phases of this research study.

7. REFERENCES

[1] M. Brambilla, J. Cabot, and M. Wimmer, "Model-driven software engineering in practice," in *Synthesis Lectures on Software Engineering*. vol. 1, ed: Morgan & Claypool, 2012.

[2] J. Hutchinson, J. Whittle, and M. Rouncefield, "Model-driven engineering practices in industry: Social, organizational and managerial factors that lead to success or failure," *Science of Computer Programming*, vol. 89, Part B, pp. 144-161, 2014.

[3] A. Gokhale, D. C. Schmidt, B. Natarajan, J. Gray, and N. Wang, "Model Driven Middleware," in *Middleware for Communications*, ed: Wiley, 2004.

[4] D. Akdur and V. Garousi, "Model-Driven Engineering in Support of Development, Test and Maintenance of Communication Middleware: An Industrial Case-Study," in *International Conference on Model-Driven Engineering and Software Development (MODELSWARD)*, France, 2015.

[5] V. Wiels, R. Delmas, D. Doose, P. L. Garoche, J. Cazin, and G. Durrieu, "Formal Verification of

Critical Aerospace Software," *AerospaceLab*, pp. p. 1-8, 2012-05-01 2012.

[6] J.-A. Maxa, M. S. Ben Mahmoud, and N. Larrieu, "1 - State of the Art of Model-driven Development (MDD) as Applied to Aeronautical Systems," in *Model-driven Development for Embedded Software*, J.-A. Maxa, M. S. Ben Mahmoud, and N. Larrieu, Eds., ed: Elsevier, 2018, pp. 1-14.

[7] D. Akdur, O. Demirörs, and V. Garousi, "Characterizing the development and usage of diagrams in embedded software systems," in *43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, Vienna, Austria, 2017.

[8] R. Heldal, P. Pelliccione, U. Eliasson, J. Lantz, J. Derehag, and J. Whittle, "Descriptive vs prescriptive models in industry," in *ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems*, France, 2016.

[9] T. Kühne, "Matters of (Meta-) Modeling," *Software & Systems Modeling*, vol. 5, pp. 369-385, 2006.

[10] A. Rodrigues da Silva, "Model-driven engineering: A survey supported by the unified conceptual model," *Computer Languages, Systems & Structures*, vol. 43, pp. 139-155, 2015.

[11] G. Kardas, "Model-driven development of multiagent systems: a survey and evaluation," *The Knowledge Engineering Review*, vol. 28, pp. 479-503, 2013.

[12] B. A. Kitchenham, T. Dybå, and M. Jørgensen, "Evidence-based software engineering," presented at the Proceedings of the 26th International Conference on Software Engineering,(ICSE’04), Washington DC, USA, 2004.

[13] B. Kitchenham, R. Pretorius, D. Budgen, O. Pearl Brereton, M. Turner, M. Niazi, *et al.*, "Systematic literature reviews in software engineering – A tertiary study," *Information and Software Technology*, vol. 52, pp. 792-805, 2010/08/01/ 2010.

[14] M. Goulão, V. Amaral, and M. Mernik, "Quality in model-driven engineering: a tertiary study," *Software Quality Journal*, vol. 24, pp. 601-633, September 01 2016.

[15] B. Kitchenham and S. Charters, "Guidelines for Performing Systematic Literature Reviews in Software engineering," Evidence Based Software Engineering Technical Report, 2007.

[16] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," presented at the Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, London, England, United Kingdom, 2014.

[17] A. Dikici, O. Turetken, and O. Demirors, "Factors influencing the understandability of process models: A systematic literature review," *Information and Software Technology*, vol. 93, pp. 112-129, 2018.

[18] D. Akdur, "Modeling Patterns and Cultures of Embedded Software Development Projects," Thesis, Doctor of Philosophy (PhD), Information Systems, Middle East Technical University (METU), www.researchgate.net/publication/322701453_Modeling_Patterns_and_Cultures_of_Embedded_Software_Development_Projects, 2018.

[19] M. Szvetits and U. Zdun, "Systematic literature review of the objectives, techniques, kinds, and architectures of models at runtime," *Software & Systems Modeling*, vol. 15, pp. 31-69, February 01 2016.

[20] D. Akdur, V. Garousi, and O. Demirörs, "A survey on modeling and model-driven engineering practices in the embedded software industry," *Journal of Systems Architecture* vol. 91, pp. 62-82, 2018.

[21] D. Akdur, V. Garousi, and O. Demirörs, "Cross-factor analysis of software modeling practices versus practitioner demographics in the embedded software industry," in *6th Mediterranean Conference on Embedded Computing (MECO)*, Montenegro, 2017.

[22] D. Akdur. (2019, Last accessed: March 28, 2019). *Online Dataset: Systematic reviews in MDE*. Available: https://www.researchgate.net/publication/337425338_Tertiary_Study_of_MDE_on_software_modeling_characteristics

[23] I. Sommerville, *Software Engineering*: Addison Wesley, 2010.

8. APPENDIX – Final Systematic Mapping

Paper ID	Paper Title	Year	Type	# of primary studies before exclusion	# of primary studies in final	Ratio of after/ before	# of paper references (if prev values are empty)
PID1	A Mapping Study on Empirical Evidence related to the Models and Forms used in the UML	2008	SM	71	33	0,46	
PID2	A survey of approaches for the visual model-driven development of next generation software-intensive systems	2006	Survey				42
PID3	A survey of model-driven testing techniques	2009	Survey				29
PID4	A survey of UML applications in mechatronic systems	2011	Survey				37
PID5	A survey of UML-based coverage criteria for software testing	2005	Survey				31
PID6	A survey on model-based testing approaches: a systematic review	2007	SLR	406	78	0,192	
PID7	A systematic identification of consistency rules for UML diagrams	2018	SM	2468	105	0,043	
PID8	A systematic literature review of use case specifications research	2015	SLR	1289	119	0,092	
PID9	A systematic literature review on the quality of uml models	2012	SLR	1500	266	0,177	
PID10	A Systematic Mapping on Model Based Testing applied to Web Systems	2013	SM	160	57	0,356	
PID11	A Systematic Mapping Study on DSL Evolution	2017	SM	98	34	0,347	
PID12	A systematic review of code generation proposals from state machine specifications	2012	SLR	3623	53	0,015	
PID13	A systematic review of empirical research on model-driven development with UML	2007	SLR	963	21	0,022	
PID14	A systematic review of model based testing tool support	2010	SLR	27	9	0,333	
PID15	A Systematic Review of Model-Based Testing in Aspect-Oriented Software Systems	2016	SLR	94	18	0,191	
PID16	A Systematic Review of Model-Driven Security	2013	SLR	10633	80	0,008	
PID17	A systematic review of the use of requirements engineering techniques in model-driven development	2010	SLR	877	65	0,074	
PID18	An extensive systematic review on the Model-Driven Development of secure systems	2015	SLR	10662	108	0,01	
PID19	Analysing the concept of quality in model-driven engineering literature: A systematic review	2014	SLR	2180	134	0,060	
PID20	Aspect-oriented model-driven code generation: A systematic mapping study	2013	SM	255	65	0,255	
PID21	Best Practices for Domain-Specific Modeling. A Systematic Mapping Study	2018	SM	143	19	0,133	
PID22	Challenges of Model-driven Modernization-An Agile Perspective	2013	SLR	43	26	0,605	
PID23	Challenges of variability in model-driven and transformational approaches: A systematic survey	2011	Survey				49
PID24	Classifying Research on UML model inconsistencies with Systematic Mapping	2013	SM	1491	198	0,13	
PID25	Consistency Rules for UML-based Domain-specific Language Models: A Literature Review	2015	SLR	5778	84	0,01	

PID26	Constraint Support in MDA Tools: A Survey	2006	Survey				34
PID27	Definitions and approaches to model quality in model-based software development – A review of literature	2009	SLR		40		
PID28	Design-Space Exploration in Model Driven Engineering - An Initial Pattern Catalogue	2014	Survey				17
PID29	Development of Critical Embedded Systems Using Model-Driven and Product Lines Techniques- A Systematic Review	2014	SLR	309	19	0,06	
PID30	Development of service-oriented architectures using model-driven development: A mapping study	2015	SLR	1962	129	0,07	
PID31	Domain-Specific Languages: A Systematic Mapping Study	2016	SM	1153	390	0,338	
PID32	Empirical evidence about the UML: a systematic literature review	2011	SLR	116	49	0,42	
PID33	Empirical studies concerning the maintenance of UML diagrams and their use in the maintenance of code: A systematic mapping study	2013	SM	808	38	0,05	
PID34	Environment modeling in model-based testing: concepts, prospects and research challenges: a systematic literature review	2015	SLR	297	61	0,21	
PID35	Execution of UML models: a systematic review of research and practice	2018	SLR	5456	82	0,015	
PID36	Extracting reusable design decisions for UML-based domain-specific languages: A multi-method study	2016	SLR	8115	84	0,010	
PID37	Formal verification of static software models in MDE: A systematic review	2014	SLR	8079	48	0,01	
PID38	Formalizing UML State Machines Semantics for Formal Analysis–A survey	2014	Survey				42
PID39	How MAD are we? Empirical evidence for model-driven agile development	2014	SLR	291	7	0,02	
PID40	Investigating the Model-Driven Development for Systems-of-Systems	2014	SLR	286	12	0,04	
PID41	MDE for BPM: a systematic review	2008	SLR	22	10	0,45	
PID42	Model Based Testing for Web Applications: A Literature Survey Presented	2016	SLR	2892	45	0,016	
PID43	Model driven web engineering: A systematic mapping study	2015	SM	2075	289	0,14	
PID44	Model-based security engineering for cyber-physical systems: A systematic mapping study	2017	SM	8814	48	0,005	
PID45	Model-based testing for software safety: a systematic mapping study	2018	SM	751	36	0,048	
PID46	Model-driven architecture based testing: A systematic literature review	2018	SLR	739	31	0,042	
PID47	Model-Driven Architecture for Cloud Applications Development, A survey	2015	Survey	95	51	0,537	
PID48	Model-Driven Engineering as a new landscape for traceability management: A systematic literature review	2012	SM	10028	29	0,003	
PID49	Model-Driven Engineering for Mobile Robot Systems: A Systematic Mapping Study	2015	SM	1681	69	0,041	
PID50	Modeling and automatic code generation for wireless sensor network applications using model-driven or business process approaches: A systematic mapping study	2017	SM	2213	77	0,035	
PID51	Research review: a systematic literature review on the quality of UML models	2011	SLR		266		
PID52	Security in model driven development: a survey	2011	SLR	2844	30	0,011	

PID53	Supporting the evolution of UML models in model driven software development: a survey	2013	Survey				159
PID54	Systematic literature review of the objectives, techniques, kinds, and architectures of models at runtime	2016	SLR	1219	242	0,199	
PID55	Systematic mapping study of model transformations for concrete problems	2016	SM	1135	82	0,072	
PID56	Systematic mapping study of template-based code generation	2018	SM	5131	481	0,094	
PID57	Systematic review of automatic test case generation by UML diagrams	2012	SLR				32
PID58	Test Case Generation from UML models-A survey	2013	Survey				31
PID59	Test case generation from UML state machine diagram: A survey	2012	Survey				32
PID60	The experimental applications of search-based techniques for model-based testing: Taxonomy and systematic literature review	2016	SLR	546	72	0,132	
PID61	Toward the tools selection in model based system engineering for embedded systems—A systematic literature review	2015	SLR	8862	61	0,007	
PID62	UML consistency rules: a systematic mapping study	2014	SM	1134	94	0,083	
PID63	UML Diagram Synthesis Techniques: A Systematic Mapping Study	2018	SM	275	14	0,051	
PID64	UML model refactoring: a systematic literature review	2013	SLR	3295	63	0,019	

9. VITAE

Deniz AKDUR is a Lead Software Engineer at ASELSAN, Inc., which is the largest Defense & Aerospace Company of Turkey. Prior to that, he worked as a Software Architect for different companies in both Turkey and United Kingdom in Consumer Electronics sector. He received his BSc degree in Computer Science from Bilkent University and MSc & PhD degrees in Information Systems from Middle East Technical University (METU), Ankara, Turkey. His specialties and research interests include software-intensive embedded systems, software engineering, software modeling, model-driven engineering, software engineering education and industry-academia collaborations.

Onur DEMİRÖRS is a Professor of Computer Engineering at the Izmir Institute of Technology (ceng.iyte.edu.tr) and the strategy director of Bilgi Grubu Ltd. (www.bg.com.tr). His current research focuses on decentralized modelling and organizational change, software measurement, and management. He has leaded major research and application projects on developing improvement and modelling techniques, on establishing and implementing modelling approaches for organizations and on establishing measurement infrastructures for software organizations. He has leaded application projects for dozens of companies to improve their processes, to establish their measurement infrastructures, to create organizational knowledge structures and to identify their software needs. He continues to teach on decentralized modelling, event based systems, software project and quality management, software measurement and innovative software development approaches.